IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of

EVANS et al

Atty. Ref.: 550-445; Confirmation No. 8224

Appl. No. 10/601,575

TC/A.U. 2181

Filed: June 24, 2003

Examiner: Lee, Chun Kuan

For: SYNCHRONISATION BETWEEN PIPELINES IN A DATA PROCESSING APPARATUS UTILIZING A SYNCHRONISATION QUEUE

* * * * * * * * * * *

March 12, 2008

Mail Stop AF
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

ARGUMENTS IN SUPPORT OF PRE-APPEAL BRIEF REQUEST FOR REVIEW

Gearty describes a *tightly-coupled* system where a coprocessor pipeline is synchronized with the main processor pipeline by passing signals with *fixed timing* from one pipeline to the other. A problem with tightly-coupled schemes is that as the length of pipeline processors increases, it is more difficult to maintain pipeline synchronization because of signal propagation delays that make it difficult to ensure signals are passed between the pipelines with the required fixed timing. Independent claims 1 and 29 solve this problem using a token-based pipeline synchronization technique where at least one synchronizing queue couples a predetermined pipeline stage in one of the pipelines with a partner pipeline stage in the other of the pipelines. The predetermined pipeline stage and the partner pipeline stage transfer tokens to achieve a *loosely-coupled* synchronization scheme.

As admitted by the Examiner, Gearty's tightly-coupled scheme lacks at least *three* features recited in the independent claims 1 and 29. The *first* is a synchronizing queue that includes a FIFO buffer with a predetermined plurality of entries. The presence of multiple

entries in the queue results in variable timing in the transfer of tokens between the predetermined pipeline stage and the partner pipeline stage. If at the time a token is placed in the queue, there are no further entries ahead of the token in the queue, then that token is transferred more quickly than if at the time the token is placed in the queue there are multiple tokens ahead of that token in the queue. The *second* missing feature is that the token includes a tag which uniquely identifies the coprocessor instruction to which the token relates. Because there is no fixed timing for the transfer of information from the predetermined pipeline stage to the partner pipeline stage, the token uniquely identifies the coprocessor instruction to which the token relates so that the partner pipeline stage can react appropriately. The *third* missing feature is that synchronization is achieved "without passing signals with fixed timing between the pipelines."

The claimed loosely-coupled synchronization scheme is flexible and can be used in situations where the tightly-coupled synchronization scheme may be inoperable, such as in systems where the length of the pipelined processors is large, where signal delay propagation makes it difficult to use a tightly coupled synchronization scheme, etc. Although the amount of slip between the pipelines is allowed to vary, the flexible loosely-coupled scheme ensures that the pipelines are correctly synchronized for crucial transfers of information.

Error #1: Martin Lacks The Claimed Synchronizing Queue FIFO Buffer

Claims 1 and 29 recite a main processor with a first pipeline, a coprocessor with a second pipeline, and a synchronizing queue coupling a predetermined pipeline stage in one of the pipelines with a partner pipeline stage in the other of the pipelines. It is this synchronizing queue which includes a FIFO buffer having a predetermined plurality of entries. Even though Martin can be coupled to a coprocessor, Martin only discusses the structure of the main MIPS R3000 RISC processor (3;10-13), which is an asynchronous processor, in contrast to the synchronous design recited in claims 1 and 29 and also as described in Gearty. Indeed, most of the

Examiner's references are to Figure 1 which illustrates the main asynchronous processor (see 4; 22-24: "FIG. 1 is a block diagram illustrating one embodiment 100 of an *asynchronous* pipeline architecture"). The FIFO structures 160a and 160b shown there are inside the main asynchronous processor 100.

The FIFO 160a is coupled between the decoder and the write back unit to store and transfer ordering information to the write back unit (2; 34-36), and the FIFO queue 160b is disposed between the program counter unit and the write back unit to store and transfer a program counter signal to the write back unit (2; 39-41). Neither FIFO couples a predetermined pipeline stage in the main processor 100 with a partner pipeline stage in a coprocessor or a predetermined pipeline stage in a coprocessor with a partner pipeline stage in the main processor.

The text referred to by the Examiner (9; 14-29) describes the need for FIFO queues 160a and b which stems from the fact that Martin's processor is *asynchronous*, i.e., a processor where each functional block of the processor only operates when data arrives, rather than the various functional blocks all being operated in accordance with a system clock signal. The FIFOs have nothing to do with synchronization between a main processor and a coprocessor.

Accordingly, nothing in Martin discloses a synchronizing queue (Martin is directed to an asynchronous design) coupling a predetermined pipeline stage in one of the pipelines (either the pipeline of the main processor or the coprocessor) with a partner pipeline stage in the other of the pipelines (either the coprocessor or the main processor). The fact that the main processor 100 may be coupled to a coprocessor does not disclose or suggest this claimed feature and has no relevance to the arguments that the Examiner seeks to make based on the teaching of Martin.

Error #2: Martin Lacks The Token Tag For A Coprocessor Instruction

Regarding the missing "tag which uniquely identifies the coprocessor instruction to which the token is related," the Examiner refers to (6; 11-23) of Martin. This text describes

fetching instructions from the instruction cache by the fetch unit, and in particular, to the standard comparison of a tag portion of a fetch address with the tag entries in the instruction cache to determine whether the requested instruction is or is not within the instruction cache, i.e., whether a hit or miss condition exists. First, the fetched instruction at (6; 11-23) is not a coprocessor instruction; rather, it is an instruction for the main processor 100. Second, the tag values held in an instruction cache are merely a portion of an instruction address and do not "uniquely" identify particular instances of an instruction. Third and even more significant, Martin's instruction cache tags have nothing to do with placing a token in a synchronizing queue between a main processor and a coprocessor. Indeed, Martin fails to describe any details of how the main processor should be coupled to a coprocessor. Nor do those instruction cache tags teach the claimed token including a *tag* which *uniquely identifies the coprocessor instruction to which the token relates*. A lookup operation performed within an instruction cache to determine whether an instruction is present in the cache is entirely unrelated to placing a token in a synchronizing queue between a processor and a coprocessor.

Error #3: Martin Lacks Synchronization Between The First And Second Pipelines Occurs Without Passing Signals With Fixed Timing Between The Pipelines

Regarding this third admitted missing feature from Gearty, the Examiner refers to (1; 41-66) of Martin. Here, Martin provides general background information on asynchronous processors. But the present claims are directed to achieving synchronization between two separate processors: a main processor and a coprocessor used to execute certain coprocessor instructions appearing in the sequence of instructions executed by the main processor. This is clear from the language of claim 1 which states that by using at least one synchronizing queue including a FIFO buffer having a predetermined plurality of entries, and by placing tokens in that queue that include a tag which uniquely identifies the coprocessor instruction to which the token

relates, the first and second pipelines (one being in the main processor and the other being in the coprocessor) are synchronized between the predetermined pipeline stage and the partner pipeline stage without passing signals with fixed timing between the pipelines. There is no teaching in Martin does not pass any signals between pipelines in different processors with fixed timing or otherwise.

Error #4: The Rationale For Modifying Gearty With Martin Is Unreasonable

The final rejection states that it would be obvious to "include Martin's asynchronous processor's FIFO and tag into Gearty's data processing apparatus for the benefit of implementing asynchronous processor having simpler architecture and faster processing speed." But that benefit only makes sense in Martin alone. Martin wants asynchronous operation—not synchronized. But Gearty's objective is synchronized operation. The modification the Examiner is proposing undermines Gearty's synchronization objective which makes it unreasonable. See *In re Gordon*. If one or both of Gearty's CPU and FPU each incorporated Martin's FIFOs so that Gearty would be asynchronous as the Examiner proposes, those FIFOs would not produce the claimed synchronization between the first and second pipelines without passing signals. In other words, the Examiner's modification moves Gearty even further away from what is claimed.

Any one of the above clear errors defeats the rejection based on Gearty and Martin. The final rejection should be withdrawn, and the application passed to allowance.

Respectfully submitted,
NIXON & VANDERHYE P.C.

By: _____
John R. Lastova
Reg. No. 33,149

901 North Glebe Road, 11th Floor
Arlington, VA 22203-1808
Telephone: (703) 816-4000